

***JOBSHEET***

# BASIS DATA LANJUT



**Jurusan Teknologi Informasi**  
**POLITEKNIK NEGERI MALANG**  
TAHUN AJARAN 2025/2026

## PERTEMUAN 1

### Pengenalan PostgreSQL

Team Teaching Basis Data Lanjut:

- Candra Bella Vista, S.Kom., MT.
- Moch Zawaruddin Abdullah, S.ST., M.Kom.
- Yan Watequlis Syaifudin, ST., MMT., PhD.
- Yoppy Yunhasnawa, S.ST., M.Sc.



Mata Kuliah : Basis Data Lanjut  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 3 (tiga)  
Pertemuan ke- : 1

## **JOBSHEET 01**

### **Pengenalan PostgreSQL**

#### **1. Review Materi Basis Data**

MySQL dan MariaDB adalah sistem manajemen basis data relasional yang memiliki banyak kesamaan. Hal ini dikarenakan kedua basis data ini bersumber pada engine yang sama.

##### **1.1. MySQL**

MySQL adalah salah satu *Relational Database Management System* (RDBMS) yang paling populer di dunia, digunakan untuk menyimpan, mengelola, dan memanipulasi data dalam bentuk tabel. MySQL bersifat *open-source* dan banyak digunakan untuk aplikasi web. Namun semenjak diakuisisi oleh Oracle, MySQL tidak sepenuhnya *open-source*, saat ini bersifat commercial license, dan hanya versi Community Edition yang masih bersifat free akan tetapi dengan keterbatasan fitur.

Fitur Utama MySQL diantaranya adalah

- Mendukung **ACID** (*Atomicity, Consistency, Isolation, Durability*) untuk transaksi.
- Memiliki beberapa *storage engine* seperti **InnoDB** (untuk transaksi) dan **MyISAM** (untuk kecepatan baca tinggi).
- Memiliki dukungan untuk *replication* (master-slave) dan *sharding*.
- Kompatibel dengan banyak bahasa pemrograman.
- Mendukung JSON, walaupun kemampuan lebih terbatas.

Kelebihan MySQL seperti

- Cepat untuk operasi *read-heavy*.
- Mudah diinstal dan digunakan.
- Banyak tutorial dan komunitas besar.
- Cocok untuk aplikasi web dengan *query* sederhana.



Kekurangan MySQL adalah

- Dukungan fitur SQL standar tidak selengkap PostgreSQL.
- Kurang optimal untuk transaksi kompleks dan data geospasial.
- Fitur JSON dan tipe data kompleks terbatas.

## 1.2. MariaDB

MariaDB merupakan RDBMS yang dibuat dari hasil fork/clonning MySQL sebelum MySQL diakuisisi Oracle. MariaDB dikembangkan untuk tetap *open-source*, dan bebas dari pengaruh lisensi komersial Oracle.

Fitur Utama MariaDB diantaranya

- Sangat kompatibel dengan MySQL (*syntax*, struktur tabel, dan *query* hampir sama), sehingga migrasi mudah.
- Memiliki beberapa *storage engine* tambahan seperti *Aria*, *ColumnStore*, dan *MyRocks*.
- Mendukung *replication*, *Cluster*, dan *multi-source replication*.
- Lebih fokus pada performa dan fitur *open-source* dibanding MySQL versi Oracle.

Kelebihan MariaDB seperti

- Open-source 100% tanpa risiko lisensi Oracle.
- Memiliki beberapa storage engine lebih banyak daripada MySQL.
- Performa baca/tulis meningkat untuk beberapa workload.
- Kompatibel dengan MySQL, sehingga aplikasi MySQL bisa berjalan di MariaDB tanpa banyak perubahan.

Kekurangan MariaDB:

- Beberapa fitur tambahan mungkin tidak sepenuhnya kompatibel dengan MySQL versi Oracle terbaru.
- Komunitas masih lebih kecil dibanding MySQL Oracle, meskipun komunitasnya tetap aktif.

- **MySQL:** Cocok untuk aplikasi web yang stabil, banyak tutorial, dan dukungan komunitas besar. Pilihan standar di banyak server produksi.
  - **MariaDB:** Alternatif open-source MySQL yang lebih bebas lisensi, lebih cepat pada beberapa kondisi, dan memiliki storage engine tambahan untuk kebutuhan khusus.

## 1.3. Data Definition Language (DDL)

*Data Definition Language* adalah bagian dari *Structured Query Language* (SQL) yang digunakan untuk **mendefinisikan struktur dan objek dalam basis data**, seperti membuat,



mengubah, atau menghapus tabel, indeks, dan skema. DDL tidak berhubungan langsung dengan manipulasi data, tetapi lebih fokus pada pengelolaan objek database itu sendiri.

Terdapat beberapa perintah utama dalam DDL seperti:

## 1. CREATE

Perintah CREATE digunakan untuk membuat objek baru di dalam database, seperti database, tabel, indeks, atau view. Contoh

- CREATE DATABASE → Perintah untuk membuat sebuah database baru

```
CREATE DATABASE db_akademik;
```

- CREATE TABLE → Perintah untuk membuat tabel baru dalam database

```
CREATE TABLE mahasiswa (
    nim INT PRIMARY KEY,
    nama VARCHAR(100),
    jurusan VARCHAR(50),
    tanggal_lahir DATE,
    jenis_kelamin ENUM('L', 'P')
);
```

- CREATE INDEX → Perintah untuk membuat indeks pada kolom sebuah tabel

```
CREATE INDEX idx_nim ON mahasiswa (nim);
```

- CREATE VIEW → Perintah untuk membuat view

```
CREATE VIEW vw_mahasiswa_perempuan AS
SELECT nim, nama, jurusan
FROM mahasiswa
WHERE jenis_kelamin = 'P';
```

## 2. ALTER

Perintah ALTER digunakan untuk mengubah struktur objek yang sudah ada, seperti tabel atau kolom. Contoh

- Perintah untuk menambahkan kolom baru

```
ALTER TABLE mahasiswa ADD alamat VARCHAR(255);
```

- Perintah untuk mengubah tipe data kolom

```
ALTER TABLE mahasiswa MODIFY COLUMN alamat TYPE TEXT;
```



- Perintah untuk menghapus kolom

```
ALTER TABLE mahasiswa DROP COLUMN alamat;
```

### 3. DROP

Perintah DROP digunakan untuk menghapus objek dari database, seperti tabel, view, atau indeks. Contoh

- DROP DATABASE → Perintah untuk menghapus database beserta seluruh objek yang ada di dalamnya

```
DROP DATABASE db_akademik;
```

- DROP TABLE → Perintah untuk menghapus tabel beserta data di dalamnya dari database

```
DROP TABLE mahasiswa;
```

- DROP INDEX → Perintah untuk menghapus indeks yang ada pada tabel

```
DROP INDEX idx_nim ON mahasiswa;
```

- DROP VIEW → Perintah untuk menghapus view yang ada pada database

```
DROP VIEW vw_mahasiswa_perempuan;
```

**DDL** adalah bagian penting dari SQL yang digunakan untuk mendefinisikan dan mengelola struktur database dan objek-objek yang ada di dalamnya, seperti tabel, kolom, indeks, dan lain-lain. Perintah-perintah DDL memungkinkan Kita untuk mendesain dan memodifikasi database sesuai kebutuhan, serta mengelola objek-objek yang ada di dalamnya.

DDL adalah tahapan awal dalam mengelola basis data sebelum melakukan proses lain seperti DML, DRL, dll.

#### 1.4. Data Manipulation Language (DML)

*Data Manipulation Language* adalah bagian dari SQL yang digunakan untuk memanipulasi data dalam tabel database. DML mencakup operasi-operasi untuk menambah (*create*), mengambil data (*retrieve/read*), mengubah (*update*), dan menghapus data (*delete*) yang ada di dalam database atau yang sering disebut dengan operasi CRUD.



Perintah-perintah dalam DML digunakan untuk bekerja dengan data yang telah ada di dalam tabel, berbeda dengan DDL (Data Definition Language) yang digunakan untuk mendefinisikan atau mengubah struktur objek dalam basis data.

Terdapat beberapa perintah utama dalam DDL seperti:

1. **INSERT** (*create / menambahkan data*)

Perintah INSERT digunakan untuk menambahkan satu atau lebih baris data ke dalam tabel. Contoh

- *Single Row Insert* → Menambahkan satu data baru pada tabel

```
INSERT INTO mahasiswa(nim, nama, jurusan, tanggal_lahir, jenis_kelamin)
VALUES (12345, 'John Doe', 'Teknik Informatika', '2000-05-10', 'L');
```

- *Multi Row Insert* → Menambahkan beberapa baris data baru sekaligus pada tabel

```
INSERT INTO mahasiswa(nim, nama, jurusan, tanggal_lahir, jenis_kelamin)
VALUES
(12346, 'Jane Smith', 'Sistem Informasi', '2001-08-15', 'P'),
(12347, 'Amin Hidayat', 'Teknik Elektro', '1999-11-20', 'L');
```

2. **SELECT** (*retrieve/read / membaca data*)

Perintah SELECT digunakan untuk mengambil data dari satu atau lebih tabel dalam database. Khusus perintah SELECT ini, bisa dimasukkan pada kategori *Data Retrieval Language* (DRL). Contoh

- Perintah untuk mengambil semua data dari satu tabel

```
SELECT * FROM mahasiswa;
```

- Perintah untuk mengambil data berdasarkan kondisi tertentu

```
SELECT * FROM mahasiswa WHERE jenis_kelamin = 'L';
```

3. **UPDATE** (*update / mengubah data*)

Perintah UPDATE digunakan untuk mengubah data yang sudah ada di dalam tabel. Perintah ini memerlukan klausa WHERE untuk memfilter baris data yang ingin diubah. Karena jika tanpa klausa WHERE, maka semua data dalam tabel akan ikut terganti, dan hal ini tentu saja tidak baik. Contoh



- Mengubah satu kolom data berdasarkan kondisi tertentu

```
UPDATE mahasiswa
SET nama = 'Johnathan Doe'
WHERE nim = 12345;
```

- Mengubah beberapa kolom data sekaligus, berdasarkan kondisi tertentu

```
UPDATE mahasiswa
SET nama = 'Jonah Saraswati',
jurusan = 'Sistem Informasi Bisnis',
tanggal_lahir = '2021-05-15',
jenis_kelamin = 'P'
WHERE nim = 12345;
```

- Query yang Tidak Disarankan (tidak ada kondisi WHERE)

```
UPDATE mahasiswa
SET nama = 'Johnathan Doe';
```

#### 4. DELETE (*delete* / menghapus data)

Perintah DELETE digunakan untuk menghapus data dari tabel berdasarkan kondisi yang diberikan dalam klausa WHERE. Tanpa klausa WHERE, perintah ini akan menghapus seluruh data dalam tabel dan hal ini tentu saja tidak baik. Contoh

- Menghapus data berdasarkan satu kondisi tertentu

```
DELETE FROM mahasiswa
WHERE nim = 123456;
```

- Menghapus data berdasarkan beberapa kondisi tertentu

```
DELETE FROM mahasiswa
WHERE jurusan = 'Teknik Informatika' AND
jenis_kelamin = 'P';
```

- Query yang Tidak Disarankan (tidak ada kondisi WHERE)

```
DELETE FROM mahasiswa;
```

#### 5. TRUNCATE (*delete* / mengosongkan isi tabel)

Perintah TRUNCATE digunakan untuk menghapus seluruh data dalam tabel, namun struktur tabel tetap ada. Ini lebih cepat daripada DELETE karena tidak memerlukan pencatatan transaksi untuk setiap baris yang dihapus. Contoh



- Menghapus data berdasarkan satu kondisi tertentu

```
TRUNCATE TABLE mahasiswa;
```

### 1.5. Data Retrieval Language (DRL)

*Data Retrieval Language* adalah bagian dari SQL yang digunakan **khusus** untuk mengambil atau membaca data dari basis data. Fokus utama DRL adalah retrieval data, sehingga DRL biasanya mencakup query SELECT beserta fitur-fitur pendukung untuk menampilkan data sesuai kebutuhan pengguna.

DRL sering dianggap sebagai bagian dari **DML**, karena **SELECT** memang termasuk operasi manipulasi data, namun DRL lebih spesifik untuk **pengambilan data saja**, bukan untuk menambah, mengubah, atau menghapus data.

Tujuan DRL diantaranya adalah

- Mengambil data dari satu atau lebih tabel dalam database.
- Menyaring data berdasarkan kondisi tertentu.
- Mengurutkan dan mengelompokkan data untuk analisis.
- Menyajikan data dalam format yang mudah dibaca.

Terdapat beberapa klausa yang digunakan dalam DRL, diantaranya

Klausa	Deskripsi
<b>SELECT</b>	Untuk mengambil data/menentukan kolom yang ingin dipilih
<b>FROM</b>	Untuk memilih sumber data yang akan diambil, bisa berupa tabel atau sub-query
<b>WHERE</b>	Untuk menunjukkan secara spesifik suatu row/baris yang akan dicari
<b>GROUP BY</b>	Untuk mengelompokkan data yang akan diambil
<b>HAVING</b>	Untuk memilih diantara kelompok data yang didefinisikan klausa GROUP BY
<b>ORDER BY</b>	Untuk mengurutkan baris data berdasarkan kolom tertentu. Nilai urutan terbagi menjadi dua, yaitu <i>asc</i> ( <i>ascending</i> ) dan <i>desc</i> ( <i>descending</i> ).
<b>AS</b>	Untuk memberikan <i>nama alias</i> ( <i>nama sementara</i> ) pada <i>object</i> seperti <i>tabel</i> , <i>kolom</i> , <i>sub-query</i>



Secara struktural, query DQL yang sesuai adalah seperti berikut

Klausa	Keterangan
<code>SELECT &lt;nama_kolom, fungsi_sql, sub_query&gt;</code>	Data yang diambil bisa berupa kolom tabel, fungsi-fungsi di SQL, maupun hasil dari <i>sub-query</i>
<code>FROM &lt;nama_tabel / sub_query&gt;</code>	Sumber data yang diambil bisa dari tabel ataupun hasil dari <i>sub-query</i>
<code>WHERE &lt;kondisi&gt;</code>	Kondisi yang digunakan bisa lebih dari satu dengan menggunakan operator ( <i>AND</i> , <i>OR</i> )
<code>GROUP BY &lt;nama_kolom&gt;</code>	Pengelompokan harus berdasarkan kolom dari tabel yang menjadi sumber data
<code>HAVING &lt;kondisi&gt;</code>	untuk mengambil kelompok (hasil dari GROUP BY) berdasarkan kondisi tertentu.
<code>ORDER BY &lt;nama_kolom&gt; [ASC   DESC]</code>	Pengurutan berdasarkan nama kolom dari sumber data secara <i>ascending/descending</i> .

## 2. PostgreSQL

PostgreSQL adalah salah satu Sistem Manajemen Basis Data Relasional (RDBMS) yang sangat populer, bersifat open-source, dan dapat digunakan untuk aplikasi dari skala kecil hingga besar. PostgreSQL mendukung hampir semua tipe data, memiliki kemampuan untuk menangani query kompleks, serta menyediakan sejumlah fitur canggih seperti:

- Dukungan ACID (*Atomicity, Consistency, Isolation, Durability*)
- *Full transaction*, yang memungkinkan rollback dan commit
- Support untuk JSON dan JSONB untuk penyimpanan data semi-terstruktur
- Ekstensi dan plugin yang memungkinkan kustomisasi, seperti PostGIS untuk data geografis
- Replikasi dan *clustering* untuk meningkatkan skalabilitas dan ketersediaan

PostgreSQL memiliki beberapa versi diantaranya mulai versi 10 sampai dengan versi 17.

Pada Mata Kuliah Basis Data Lanjut ini, kita akan menggunakan  
**PostgreSQL versi 15**



## 2.1. Perbedaan PostgreSQL dan MySQL

MySQL dan PostgreSQL adalah dua Sistem Manajemen Basis Data Relasional (RDBMS) yang sangat populer dan sering digunakan dalam pengembangan aplikasi. Meski keduanya berfungsi sebagai alat untuk mengelola basis data relasional, ada beberapa perbedaan mendasar antara MySQL dan PostgreSQL yang dapat memengaruhi pilihan pengembang berdasarkan kebutuhan spesifik.

Walaupun sama-sama RDBMS, PostgreSQL dan MySQL memiliki beberapa perbedaan, diantaranya

### 1. Lisensi

PostgreSQL dan MySQL memiliki perbedaan lisensi, jika PostgreSQL mengusung full *open-source*, lain hanya dengan MySQL yang ada *commercial license* dari Oracle selaku pemegang nama MySQL saat ini.

MySQL	PostgreSQL
MySQL menggunakan lisensi GPL (General Public License), yang memungkinkan pengguna untuk menggunakaninya secara bebas. Namun, setelah diakuisisi oleh Oracle, MySQL sering kali dianggap memiliki batasan tertentu di dalam ekosistem komersial.	PostgreSQL menggunakan lisensi PostgreSQL License, yang mirip dengan MIT. Ini berarti PostgreSQL sepenuhnya <i>open-source</i> dan dapat digunakan tanpa batasan lisensi dalam proyek komersial atau pribadi.

### 2. Fungsionalitas dan Fitur

MySQL	PostgreSQL
<ul style="list-style-type: none"><li>- Kinerja tinggi untuk operasi sederhana: MySQL sangat cepat untuk operasi dasar seperti <i>INSERT</i>, <i>SELECT</i>, dan <i>UPDATE</i>, terutama dengan penggunaan engine <i>InnoDB</i>.</li><li>- Lebih cocok untuk aplikasi web: MySQL lebih sering digunakan untuk aplikasi web ringan hingga menengah dengan kebutuhan database yang relatif sederhana.</li><li>- Tidak memiliki dukungan lengkap untuk operasi kompleks: MySQL memiliki dukungan terbatas untuk beberapa fitur lanjutan seperti <i>window functions</i> dan <i>common table expressions</i></li></ul>	<ul style="list-style-type: none"><li>- Lebih canggih dan kaya fitur: PostgreSQL memiliki fitur yang lebih kaya dan lebih canggih, seperti <i>ACID compliance</i>, <i>window functions</i>, <i>common table expressions</i> (CTE), <i>advanced indexing</i>, dan <i>PostGIS</i>.</li><li>- Dukungan untuk tipe data kompleks: PostgreSQL mendukung berbagai tipe data kompleks seperti <i>JSON</i>, <i>XML</i>, <i>arrays</i>, <i>hstore</i>, dan <i>user-defined types</i>.</li><li>- Ekstensi dan kustomisasi: PostgreSQL memungkinkan penggunaan ekstensi dan plugin, yang membuatnya lebih fleksibel untuk penggunaan tertentu seperti <i>PostGIS</i> (untuk data geospasial).</li></ul>



(CTE) (meskipun fitur ini terus berkembang).	
--	--

### 3. Kinerja

PostgreSQL dan MySQL memiliki tingkat kinerja yang berbeda, sesuai dengan peruntukannya

MySQL	PostgreSQL
<ul style="list-style-type: none"><li>- Lebih cepat untuk aplikasi sederhana: MySQL sering kali lebih cepat dalam operasi dasar, terutama dalam aplikasi web dengan tingkat <i>read-heavy</i>.</li><li>- Penggunaan InnoDB untuk transaksi dan kinerja lebih baik pada aplikasi yang menuntut penulisan data.</li><li>- Tidak seefisien PostgreSQL untuk operasi yang sangat kompleks.</li></ul>	<ul style="list-style-type: none"><li>- Optimal untuk transaksi kompleks: PostgreSQL sangat baik untuk aplikasi yang memerlukan query kompleks, analisis data yang dalam, dan write-heavy workloads.</li><li>- Kinerja tinggi untuk aplikasi yang membutuhkan integritas data tinggi dan konsistensi transaksi.</li></ul>

### 4. Dukungan Transaksi

Baik PostgreSQL maupun MySQL mendukung proses transaksional akan tetapi berbeda dalam prosesnya

MySQL	PostgreSQL
<ul style="list-style-type: none"><li>- Dukungan transaksi terbatas pada engine: MySQL menawarkan dukungan transaksi dengan engine InnoDB, yang memberikan fitur ACID untuk transaksinya.</li><li>- Tidak memiliki dukungan penuh untuk transaksi kompleks dibandingkan PostgreSQL.</li></ul>	<ul style="list-style-type: none"><li>- Transaksi penuh dengan ACID: PostgreSQL mendukung transaksi sepenuhnya dengan memastikan Atomicity, Consistency, Isolation, dan Durability (ACID).</li><li>- Lebih kuat dalam hal transaksi dan konsistensi data, dengan dukungan untuk <i>nested transactions</i> menggunakan <i>SAVEPOINT</i>.</li></ul>

### 5. Kompatibilitas dan Standar SQL

PostgreSQL dan MySQL memiliki kebijakan sendiri-sendiri dalam melakukan standard query SQL yang diterapkan

MySQL	PostgreSQL
<ul style="list-style-type: none"><li>- Lebih fleksibel dengan sintaks SQL: MySQL lebih fleksibel dalam beberapa hal, seperti pemrosesan string dan pengelolaan indeks.</li></ul>	<ul style="list-style-type: none"><li>- Kepatuhan lebih tinggi terhadap standar SQL: PostgreSQL lebih ketat dalam mengikuti standar SQL dan mendukung fitur SQL standar dengan lebih baik.</li></ul>



- Fitur yang tidak sepenuhnya sesuai dengan standar SQL: MySQL menggunakan cara tersendiri dalam mengelola autoincrement dan pembatasan foreign key.	- Dukungan penuh untuk tipe data standar seperti ENUM, ARRAY, dan JSON.
--	---

MySQL lebih cocok untuk aplikasi web dengan kecepatan tinggi dan struktur data sederhana, serta penggunaan read-heavy workloads. PostgreSQL lebih cocok untuk aplikasi enterprise dengan kebutuhan transaksi kompleks, integritas data tinggi, dan fungsionalitas lanjutan seperti analisis data geospasial, JSON, dan query kompleks. Pemilihan antara MySQL dan PostgreSQL sangat bergantung pada kebutuhan aplikasi dan skala aplikasi yang dikembangkan.

Jika proyek membutuhkan keandalan transaksi, kompleksitas query, dan ekstensi khusus, maka PostgreSQL adalah pilihan yang tepat.

## 2.2. Perangkat Pendukung untuk PostgreSQL

Perangkat lunak pendukung untuk PostgreSQL adalah aplikasi atau *tools* yang digunakan bersama PostgreSQL untuk mempermudah administrasi, pengelolaan, pengembangan, monitoring, maupun analisis data. *Tools* ini bisa berupa GUI, command-line utilities, driver, dan ekstensi tambahan yang membantu bekerja dengan PostgreSQL.

Berikut beberapa tools yang mempermudah dalam mengelola data pada PostgreSQL

### 1. pgAdmin

pgAdmin adalah alat manajemen basis data grafis (GUI) resmi untuk PostgreSQL, yang memungkinkan pengguna untuk mengelola, memonitor, dan mengoperasikan database PostgreSQL dengan cara yang lebih mudah dan visual dibandingkan menggunakan baris perintah (psql).

pgAdmin digunakan untuk melakukan hampir semua tugas administrasi dan pengelolaan database, termasuk:

- Membuat database, tabel, dan kolom.
- Menulis dan mengeksekusi query SQL.
- Mengatur user dan hak akses.
- Memantau performa server.
- Mengelola backup dan restore database

pgAdmin tersedia dalam versi desktop dan versi web.



Beberapa kelebihan pgAdmin diantaranya:

- Memudahkan pengguna baru untuk menggunakan PostgreSQL tanpa harus menghafal perintah SQL.
- Menyediakan visualisasi dan tool monitoring yang lengkap.
- Mendukung multiplatform dan server lokal maupun remote.
- Mempercepat administrasi database besar atau kompleks.

Kekurangan dari pgAdmin diantaranya

- Bisa terasa lebih lambat untuk database sangat besar dibanding menggunakan psql.
- Mengandalkan GUI, sehingga beberapa operasi kompleks masih lebih cepat dilakukan lewat command-line.
- Memerlukan instalasi tambahan di beberapa sistem (desktop/web server).

## 2. DBeaver

DBeaver adalah perangkat lunak manajemen basis data universal (*universal database management tool*) yang bersifat *cross-platform* dan mendukung banyak jenis Database Management System (DBMS) termasuk PostgreSQL, MySQL, MariaDB, Oracle, SQL Server, SQLite, dan banyak lagi. DBeaver sangat populer di kalangan developer dan administrator database karena kemampuannya yang lengkap dan GUI yang user-friendly.

Beberapa kelebihan DBeaver adalah:

- Multi-platform: Bisa dijalankan di Windows, macOS, dan Linux.
- Support banyak DBMS: Tidak terbatas pada satu jenis database.
- User-friendly GUI: Memudahkan pemula sekalipun untuk mengelola database.
- Open-source: DBeaver Community Edition gratis digunakan.
- Fitur lengkap: Query editor, ERD, import/export, monitoring, dan administrasi database.

Kekurangan dari DBeaver :

- Memerlukan resource cukup tinggi: Untuk database besar, GUI bisa terasa lambat.
- Versi Enterprise berbayar: Fitur tambahan seperti support cloud dan advanced analytics memerlukan lisensi.
- Beberapa fitur lanjutan sedikit lebih kompleks dibanding tools spesifik DBMS, seperti pgAdmin untuk PostgreSQL.



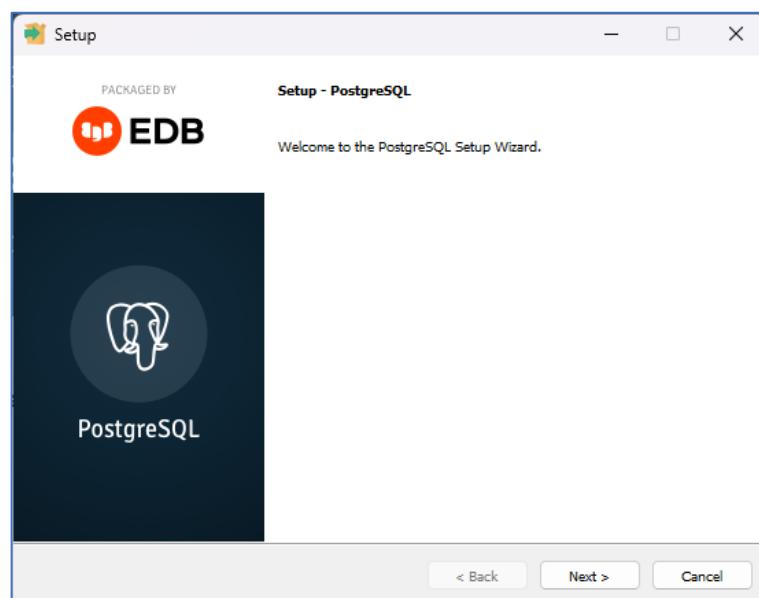
## Praktikum 01 – Install PostgreSQL

1. Kita download PostgreSQL Versi 15 di

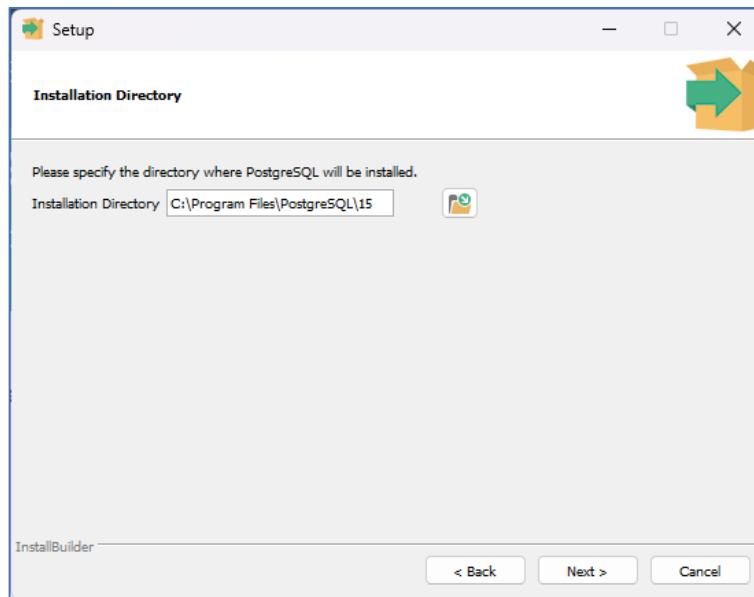
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
17.6	<a href="#">postgresql.org</a>	<a href="#">postgresql.org</a>			Not supported
16.10	<a href="#">postgresql.org</a>	<a href="#">postgresql.org</a>			Not supported
15.14	<a href="#">postgresql.org</a>	<a href="#">postgresql.org</a>			Not supported
14.19	<a href="#">postgresql.org</a>	<a href="#">postgresql.org</a>			Not supported

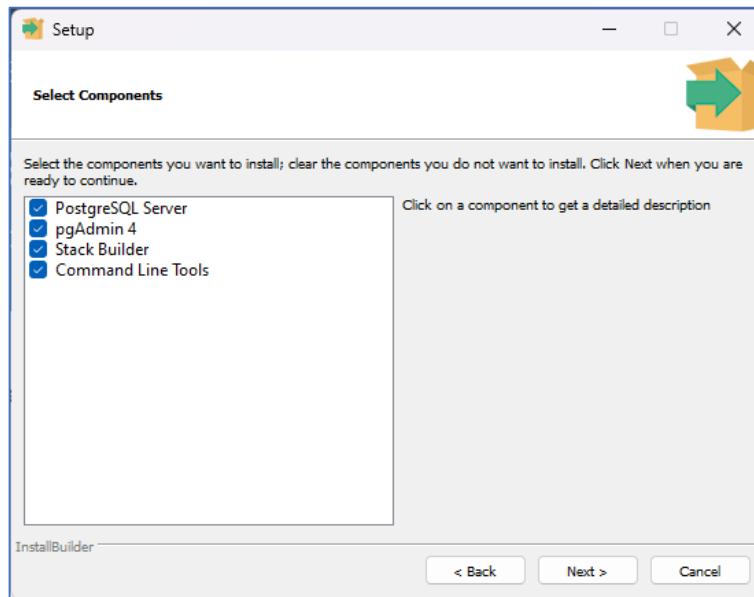
2. Silahkan sesuaikan dengan Sistem Operasi pada laptop kalian masing-masing. Contoh disini kita menggunakan Sistem Operasi Windows 11 64bit
3. Selanjutnya kita jalankan file *installer* PostgreSQL nya



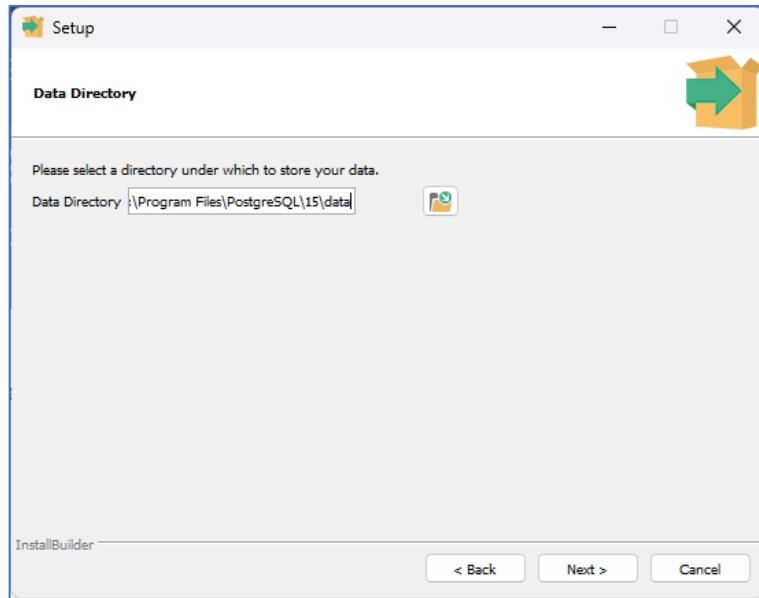
4. Pilih direktori tempat database PostgreSQL akan di-*install*



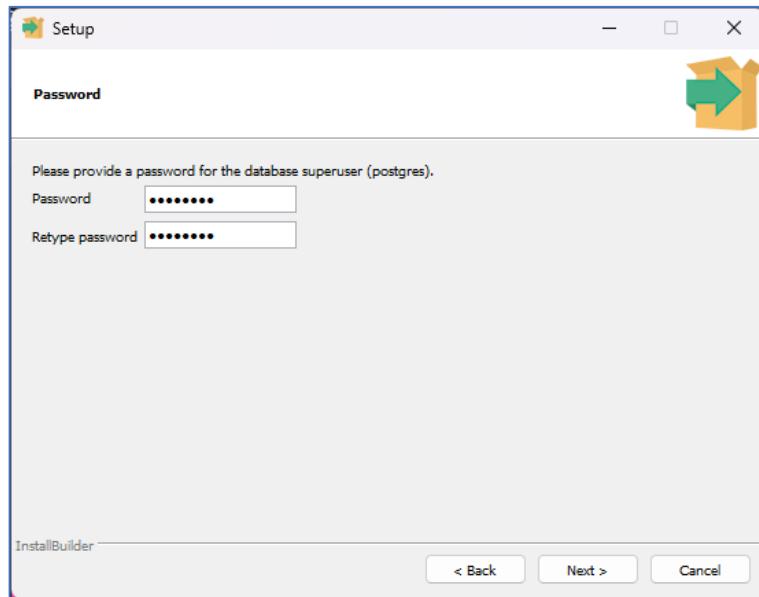
5. Selanjutnya, kita centang semua opsi untuk meng-*install* PostgreSQL beserta perangkat lunak pendukung (termasuk pgAdmin)



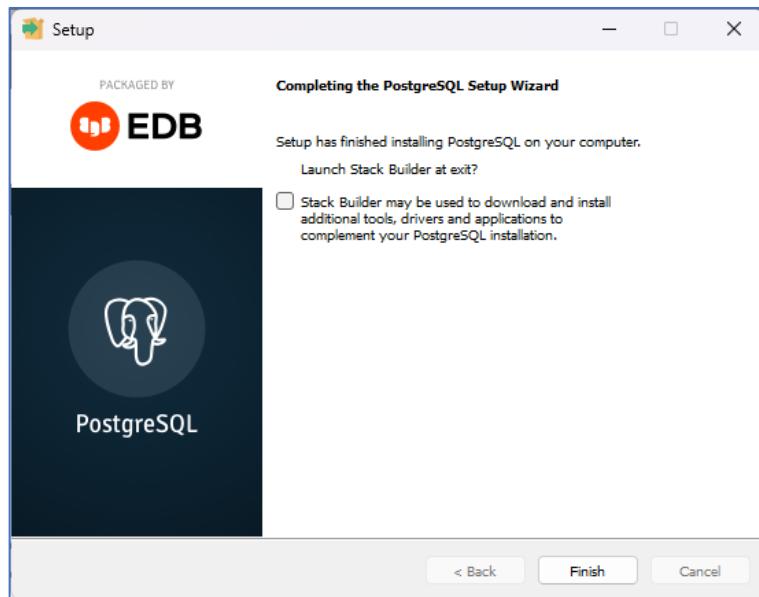
6. Selanjutnya, pilih lokasi direktori untuk menyimpan data dari database PostgreSQL. Kita bisa memilih opsi default dari *installer*



7. Selanjutnya kita set **default password** untuk user **postgres**. (contoh password: 12345678)



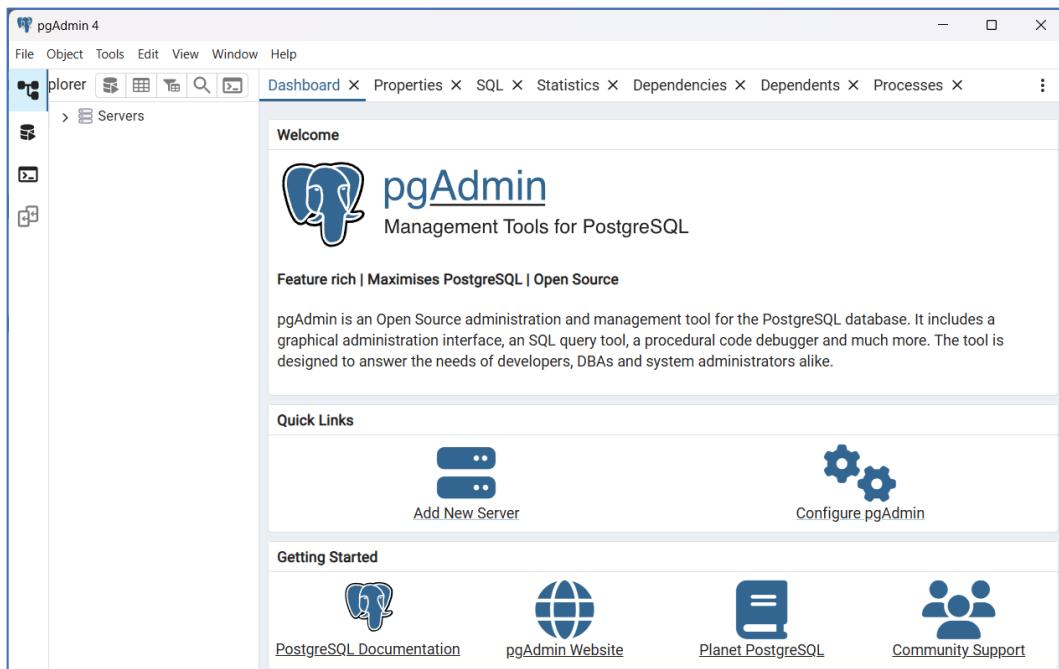
8. Kita set default port dari PostgreSQL, biasanya adalah port 5432
9. Kita buat config **locale** ke **default** saja
10. Selanjutnya klik next sampai proses instalasi selesai
11. Diakhir instalasi akan tampil seperti gambar berikut, kita bisa uncheck untuk skip dulu menjalankan *Stack Builder* dari PostgreSQL.



12. Silahkan cek apakah instalasi PostgreSQL beserta pgAdmin sudah berhasil di-install pada laptop kalian masing-masing.

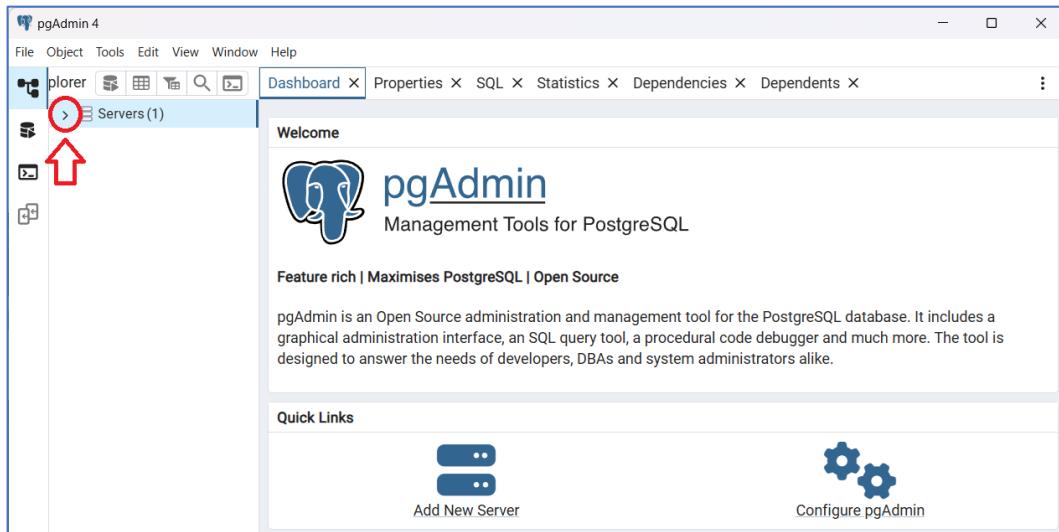
### Praktikum 02 – Check PostgreSQL dan pgAdmin

1. Setelah kita berhasil install PostgreSQL dan pgAdmin, kita bisa buka pgAdmin untuk proses setting koneksi ke PostgreSQL nya. Berikut tampilan awal pgAdmin setelah dijalankan

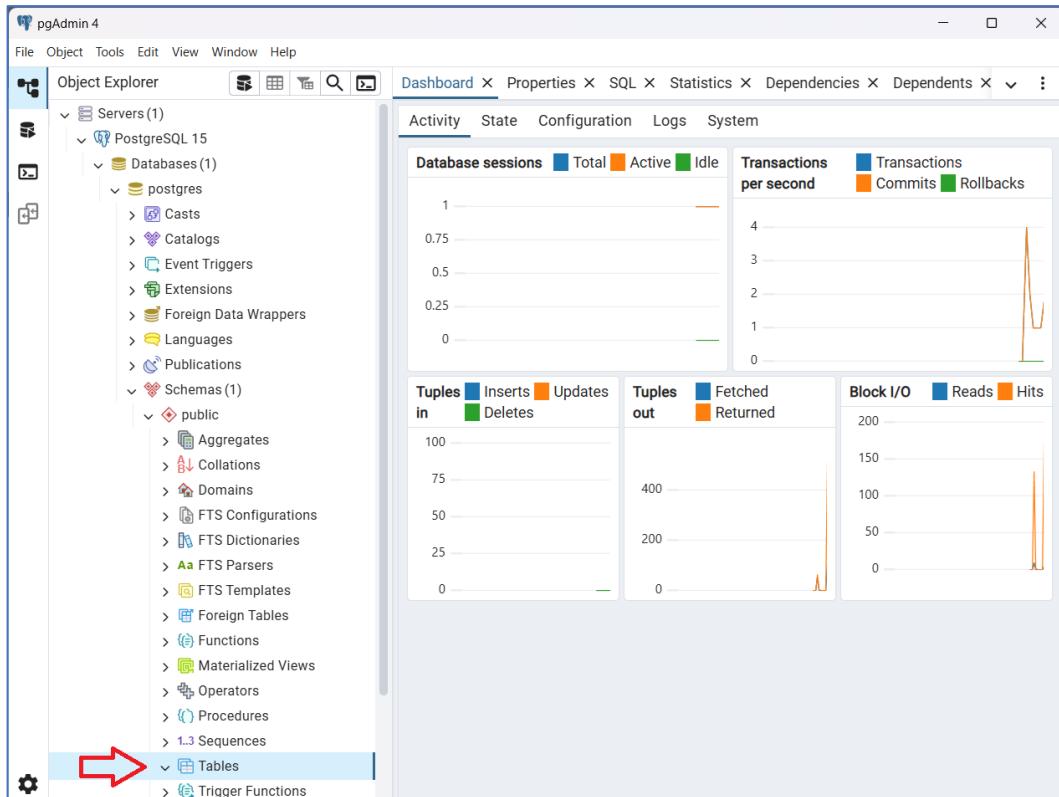




2. Kita klik tanda sesuai panah untuk membuka server lokal PostgreSQL yang ada di laptop kita



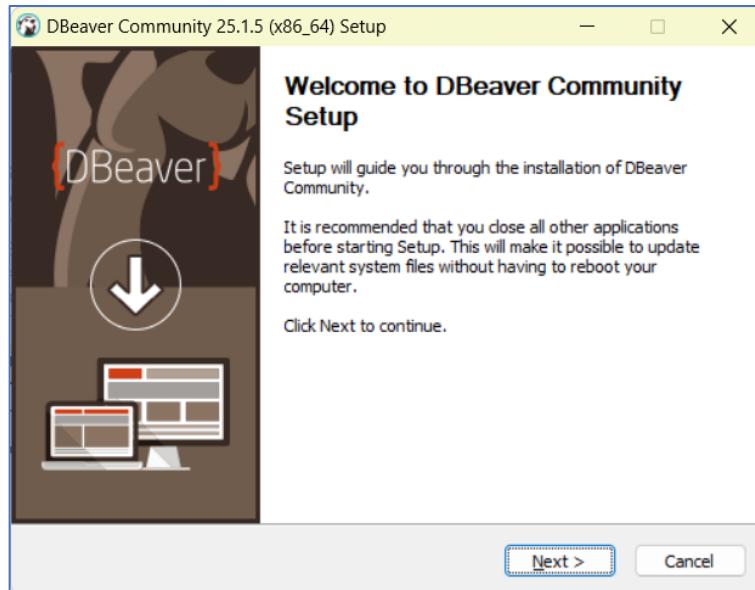
3. Setelah itu akan muncul popup untuk memasukkan password untuk masuk ke PostgreSQL. Kita masukkan password (contoh 12345678) sesuai saat instalasi
4. Setelah berhasil masuk ke PostgreSQL, secara default, kita bisa melihat tabel-tabel di Postgre pada **Databases → postgres → Schemas → public → Tables**



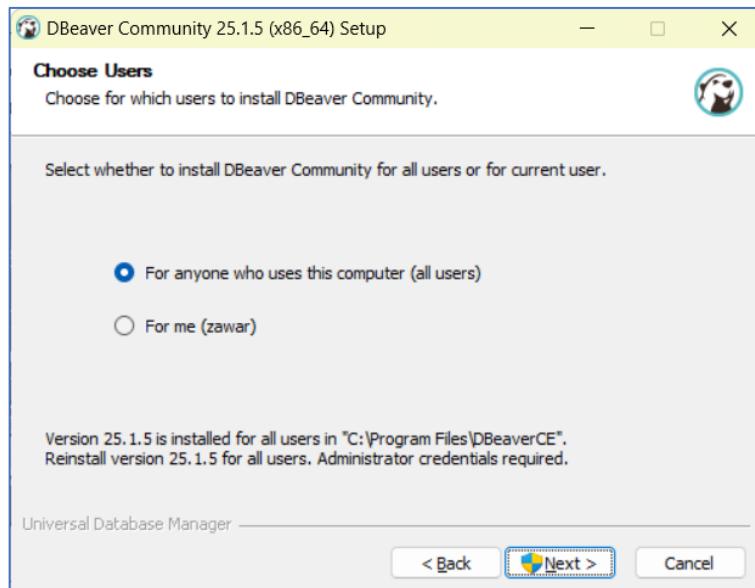


## Praktikum 03 – Install DBeaver dan Menghubungkan ke PostgreSQL

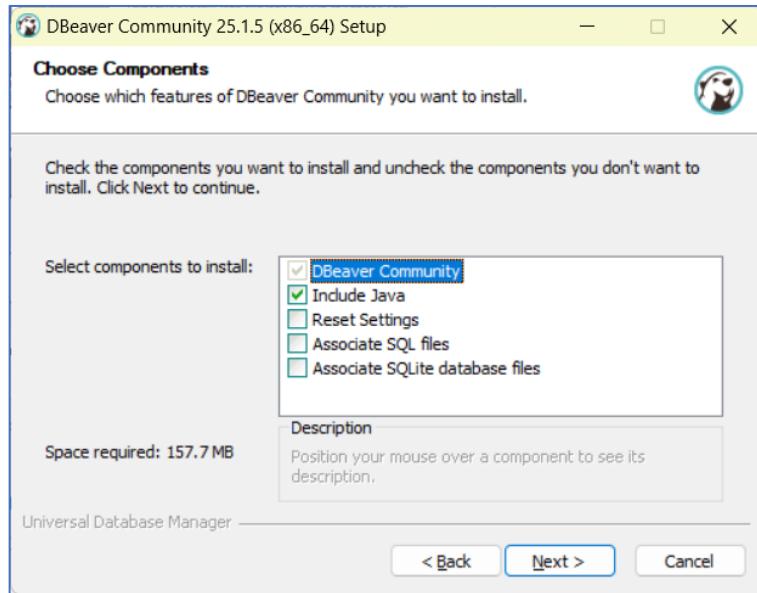
1. DBeaver bisa kita download disini <https://dbeaver.io/download/>, kita download yang versi *Community Edition*
2. Kita jalankan installer DBeaver



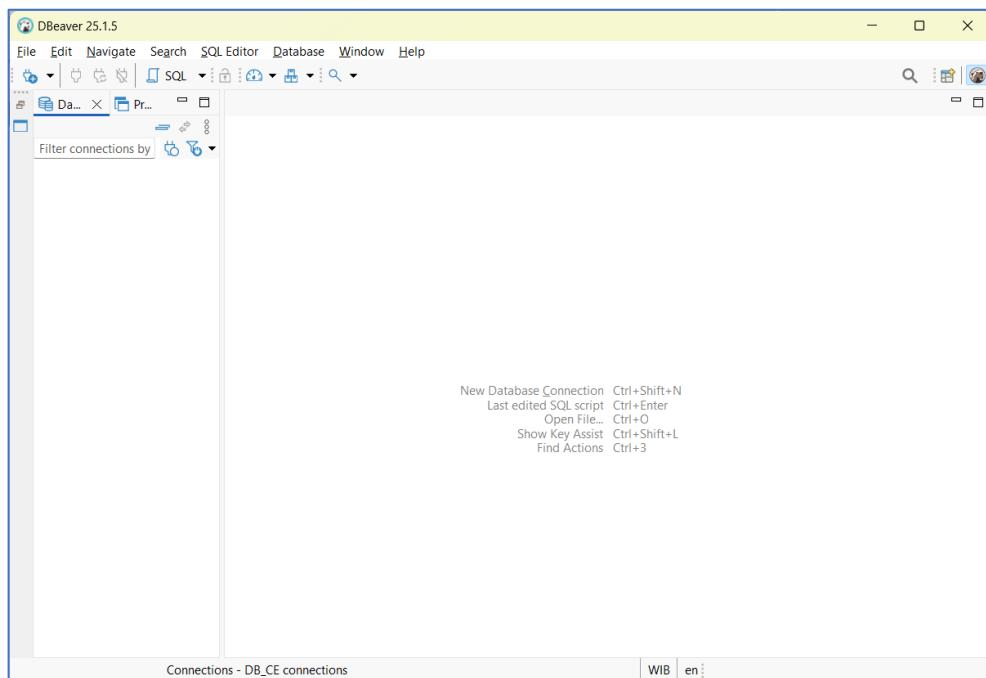
3. Kita bisa pilih opsi agar DBeaver bisa digunakan oleh semua user di laptop kita



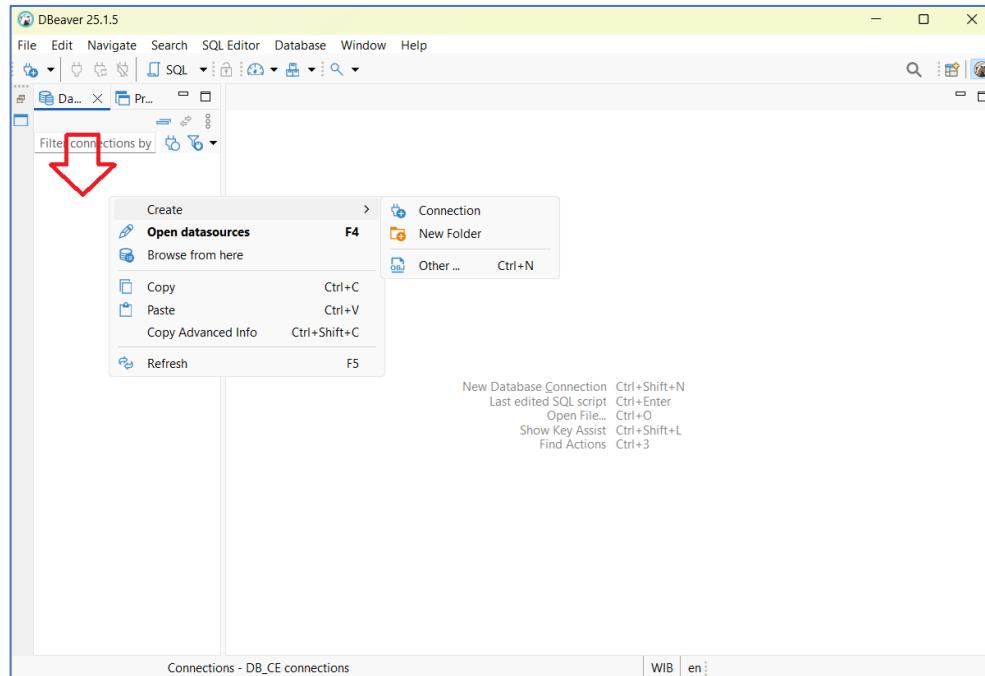
4. Cukup kita centang dua opsi atas



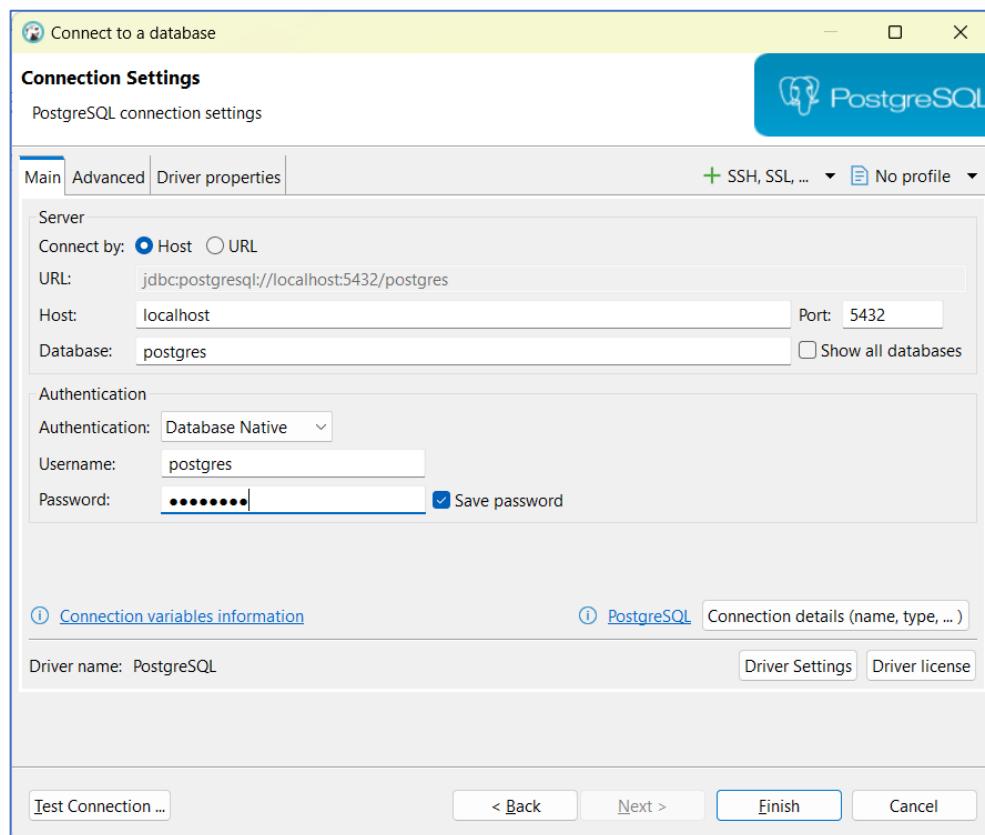
5. Selanjutnya kita pilih opsi default DBeaver sampai instalasi berhasil
6. Setelah DBeaver berhasil di-install, kita buka aplikasi DBeaver



7. Kemudian kita buat koneksi ke PostgreSQL, dengan cara *klik kanan pada bagian seperti pada Database Navigator → Create → Connection*

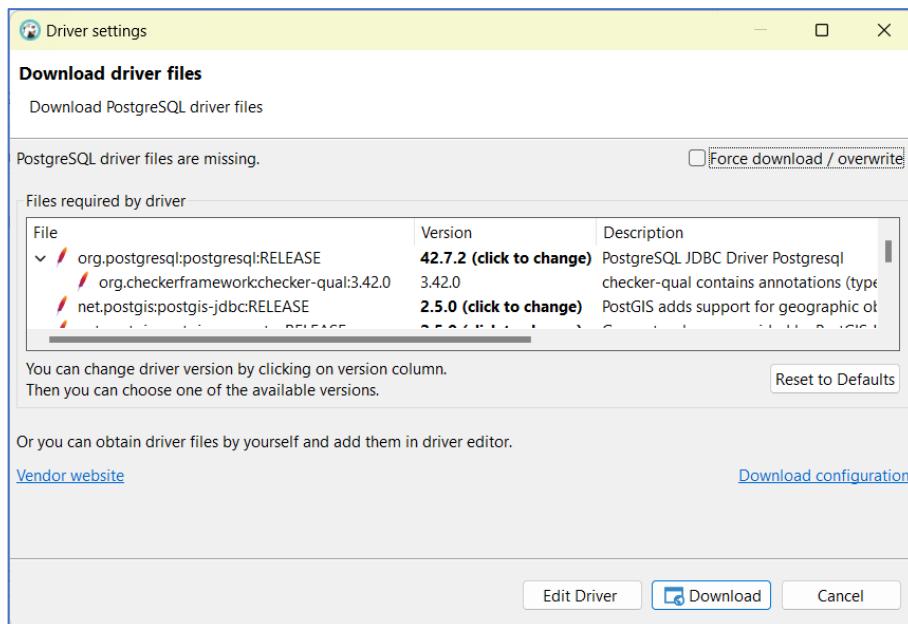


8. Selanjutnya kita pilih Database PostgreSQL dan klik next
9. Kemudian kita ini pengaturan koneksi database

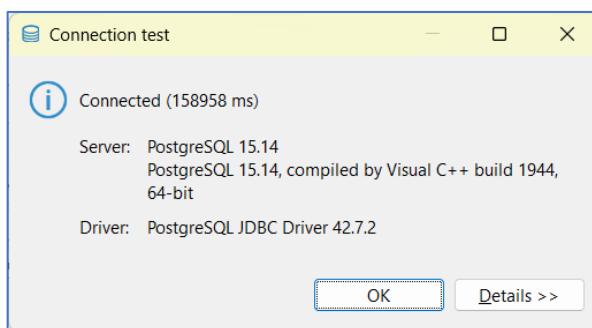




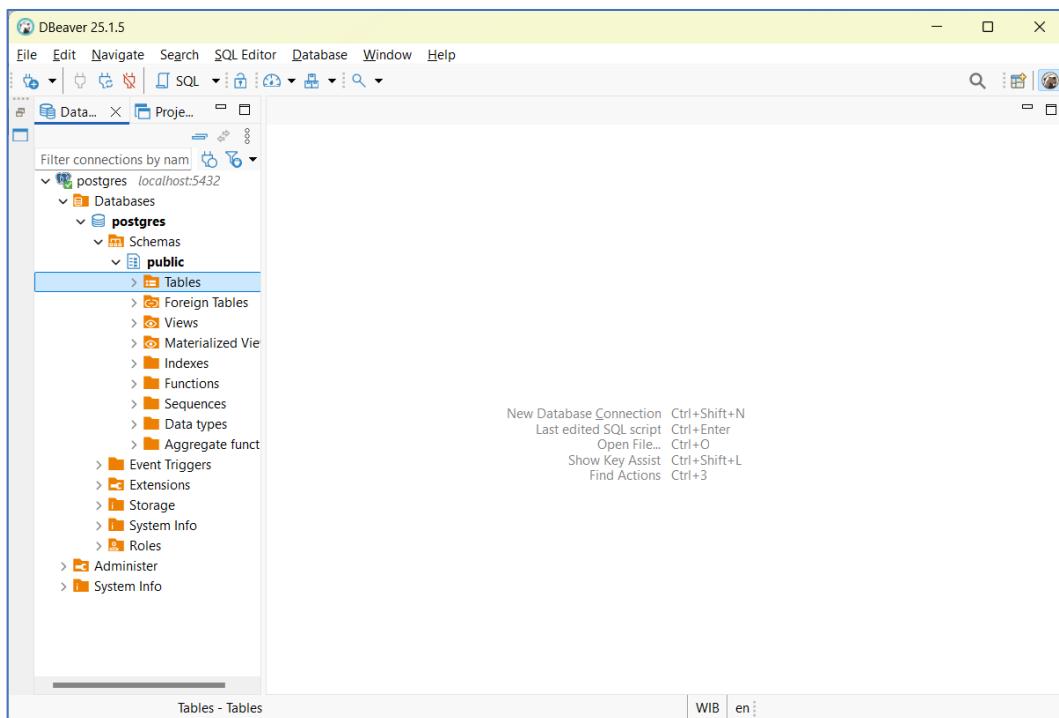
10. Kita bisa klik tombol **Test Connection**, pada bagian kiri bawah. Hal ini dilakukan untuk mengecek apakah pengaturan koneksi kita sudah sesuai dan driver untuk Database yang dituju sudah ready.
11. Jika muncul popup seperti ini, maka dibutuhkan driver untuk bisa terhubung ke database PostgreSQL. Kita klik download pada popup tersebut



12. Setelah melakukan download driver berhasil, maka proses test connection akan dilakukan
13. Opsi koneksi berhasil, maka akan tampil popup seperti berikut



14. Berikut adalah tampilan jika kita aplikasi DBeaver berhasil terkoneksi dengan database PostgreSQL



\*\*\* Sekian, dan selamat belajar \*\*\*